Perfect Sampling of $GI/GI/c$ Queues

Blanchet, J., Dong, J., and Pei, Y.

Abstract

We introduce the first class of perfect sampling algorithms for the steady-state distribution of multi-server queues with general interarrival time and service time distributions. Our algorithm is built on the classical dominated coupling from the past protocol. In particular, we use a coupled multi-server vacation system as the upper bound process and develop an algorithm to simulate the vacation system backwards in time from stationarity at time zero. The algorithm has finite expected termination time with mild moment assumptions on the interarrival time and service time distributions.

1 Introduction

In this paper, we present the first class of perfect sampling algorithms for the steady-state distribution of multi-server queues with general interarrival time and service time distributions. Our algorithm has finite expected running time under the assumption that the interarrival times and service times have finite $2+\epsilon$ moment for some $\epsilon > 0$.

The goal of perfect sampling is to sample without any bias from the steady-state distribution of a given ergodic process. The most popular perfect sampling protocol, known as Coupling From The Past (CFTP), was introduced by Propp and Wilson in the seminal paper [17]; see also [2] for another important early reference on perfect simulation.

Foss and Tweedie [11] proved that CFTP can be applied if and only if the underlying process is uniformly ergodic, which is not a property applicable to multi-server queues. So, we use a variation of the CFTP protocol called Dominated CFTP (DCFTP) introduced by Kendall in [16] and later extended in [15, 14].

A typical implementation of DCFTP requires at least four ingredients:

a) a stationary upper bound process for the target process,

b) a stationary lower bound process for the target process,

c) the ability to simulate a) and b) backwards in time (i.e. from time $[-t, 0]$ for any $t > 0$)

d) a finite time $-T < 0$ at which the state of the target process is determined (typically by having the upper and lower bounds coalesce), and the ability to reconstruct the target process from $-T$ up to time 0.

The time $-T$ is called the coalescence time and it is desirable to have $E(T) < \infty$. The ingredients are typically combined as follows. One simulates a) and b) backwards in time (by applying c)) until the processes meet. The target process is sandwiched between a) and b). Therefore, if we can find a time $-T < 0$ when processes a) and b) coincide, the state of the target process is known at $-T$ as well. Then, applying d), we reconstruct the target process from $-T$ up to time 0. The algorithm outputs the state of the target process at time 0.

It is quite intuitive that the output of the above construction is stationary. Specifically, assume that the sample path of the target process, coupled with, a) and b) is given from $(-\infty, 0]$, Then we can think of the simulation procedure in c) as simply observing or unveiling the paths of a) and b) during $[-t, 0]$. When we find a time $-T < 0$ at which the paths of a) and b) take the same value, because of the sandwiching property, the target process must share this common value at $-T$. Starting from that point, property d) simply unveils the path of the target process. Since this path has been coming from the infinite distant past (we simply observed it from time $-T$), the output is stationary at time 0.

One can often improve the performance of a DCFTP protocol if the underlying target process is monotone [14], as in the multi-server queue setting. A process is monotone if there exists a certain partial order, $\preceq$, such that if w and w' are initial states where $w \preceq w'$, and one uses common random numbers to simulate two paths, one starting from w and the other from w', then the order is preserved when comparing the states of these two paths at any point in time. Thus, instead of using the bounds a) and b) directly to detect coalescence, one could apply monotonicity to detect coalescence as follows. At any time $-t < 0$, we can start two paths of the target process, one from the state w' obtained from the upper bound a) observed at time $-t$, and the other from the state $w \preceq w'$ obtained from the lower bound b) observed at time $-t$. Then we run these two paths using the common random numbers, which are consistent with the backwards simulation of a) and b), in reverse order according to the dynamics of the target process, and check if these two paths meet before time zero. If they do, the coalescence occurs at such meeting time. We also notice that because we are using common random numbers and system dynamics, these two paths will merge into a single path from the coalescence time forward, and the state at time zero will be the desired stationary draw. If coalescence does not occur, then one can simply let $t \longleftarrow 2t$, and repeat the above procedure. For this iterative search procedure, we must show that the search terminates in finite time.

While the DCFTP protocol is relatively easy to understand, its application is not straightforward. In most applications, the most difficult part has to do with element c). Then, there is the issue of finding good bounding processes (elements a) and b)), in the sense of having short coalescence times - which we interpret as making sure that $E(T) < \infty$. There has been a substantial amount of research which develops generic algorithms for Markov chains (see for example [9] and [7]). These methods rely on having access to the transition kernels which is difficult to obtain in our case. Perfect simulation for queueing systems has also received significant amount of attention in recent years, though most perfect simulation algorithms for queues impose Poisson assumptions on the arrival process. Sigman [19, 20] applied the DCFTP and regenerative idea to develop perfect sampling algorithms for stable $M/G/c$ queues. The algorithm in [19] requires the system to be super-stable (i.e. the system can be dominated by a stable $M/G/1$ with). The algorithm in [20] works under natural stability conditions, but it has infinite expected termination time. A recent work by Connor and Kendall [8] extends Sigman's algorithm [20] to sample stationary $M/G/c$ queues and the algorithm has finite expected termination time, but they still require the arrivals to be Poisson. The main reason for the Poisson arrival assumption is that under this assumption one can find dominating systems which are quasi-reversible (see Chapter 3 of [13]) and therefore can be simulated backwards in time using standard Markov chain constructions (element c)).

For general renewal arrival process, our work is close in the spirit to [10], [3], [4] and [5], but the model treated is fundamentally different Thus, it requires some new developments. We also use a different coupling construction to that introduced in [20] and refined in [8]. In particular, we take advantage of a vacation system which allows us to transform the problem into simulating the running infinite horizon maximum (from time t to infinity) of renewal processes, compensated with a negative drift so that the infinite horizon maximum is well defined. Finally, we note that a significant advantage of our method, in contrast to [20] is that we do not need to empty the system in order to achieve coalescence. This is important in many server queues in heavy traffic for which it would take an exponential amount of time (in the arrival rate) or sometimes impossible to observe an empty system.

The rest of the paper is organized as follows. In Section 2 we describe our simulation strategy, involving elements a) to d), and we conclude the section with the statement of a result which summarizes our main contribution (Theorem 1). Subsequent sections (Section 3, 4 & 5) provide more detailed justification for our simulation strategy. Lastly we conduct numerical experiments in Section 6. An online companion of this paper includes a Matlab implementation of the algorithm.

2 Simulation Strategy and Main Result

Our target process is the stationary process generated by a multi-server queue with independent and identically distributed (iid) interarrival times and iid service times which are independent of the arrivals. There are $c \geq 1$ identical servers, each can serve at most one customer at a time. Customers are served on a first-come-first-served (FCFS) basis. Let $G(\cdot)$ and $\bar{G}(\cdot) = 1 - G(\cdot)$ (resp. $F(\cdot)$ and $\bar{F}(\cdot) = 1 - F(\cdot)$) denote the cumulative distribution function, CDF, and the tail CDF of the interarrival times (resp. service times). We shall use A to denote a random variable with CDF G, and V to denote a random variable with CDF F.

Assumption 1 (A1)**.** *Both A and V are strictly positive with probability one and there exists $\epsilon > 0$, such that*

$$E[A^{2+\epsilon}] < \infty, \quad E[V^{2+\epsilon}] < \infty.$$

The previous assumption will allow us to conclude that the coalescence time of our algorithm has finite expectation. The algorithm will terminate with probability one if $E[A^{1+\epsilon}] + E[V^{1+\epsilon}] < \infty$. The assumption of A and V being strictly positive can be done without loss of generality because one can always reparametrize the input in cases where either A or V has an atom at zero.

We assume that $G(\cdot)$ and $F(\cdot)$ are known so that the required parameters in Section 3.1.1 of [5] can be obtained. We write $\lambda = (\int_0^\infty \bar{G}(t)dt)^{-1} = 1/E[A]$ as the arrival rate, and $\mu = (\int_0^\infty \bar{F}(t)dt)^{-1} = 1/E[V]$ as the service rate. In order to ensure the existence of the stationary distribution of the system, we require the following stability condition $\lambda/(c\mu) < 1$.

2.1 Elements of the simulation strategy: upper bound and coupling

We first introduce some additional notations which we shall use to describe the upper bound a) in the application of the DCFTP framework. Let

$$\mathcal{T}^0 := \{T_n^0 : n \in \mathbb{Z}\backslash\{0\}\}$$

be a time-stationary renewal point process with $T_n^0 > 0$ if $n \geq 1$ and $T_{-n} < 0$ if $n \geq 1$ (the T_n^0's are sorted in a non-decreasing order in n). The time T_n^0 for $n \geq 1$ represents the arrival time of the n-th customer into the system after time zero and, for $n \geq 1$, T_{-n}^0 is the arrival time of the n-th customer, backwards in time, from time zero. We also define $T_n^{0,+} = \inf\{T_m^0 : T_m^0 > T_n^0\}$, that is, the arrival time of the next customer after T_n^0. If $n \geq 1$ or $n \leq -2$, $T_n^{0,+} = T_{n+1}^0$. However, $T_{-1}^{0,+} = T_1^0$. Similarly, we write $T_n^{0,-} = \sup\{T_m^0 : T_m^0 < T_n^0\}$.

Define $A_n = T_n^{0,+} - T_n^0$ for all $n \in \mathbb{Z}\backslash\{0\}$. Note that A_n is the interarrival time between the customer arriving at time T_n^0 and the next customer. A_n has CDF $G(\cdot)$ for $n \geq 1$ or $n \leq -2$, but A_{-1} has a different distribution due the inspection paradox.

Now, for $i \in \{1, 2, ..., c\}$ we introduce iid time-stationary renewal point processes

$$\mathcal{T}^i := \{T_n^i : n \in \mathbb{Z}\backslash\{0\}\},$$

as before we have that $T_n^i > 0$ for $n \geq 1$ and $T_{-n}^i < 0$ if $n \geq 1$ with the T_n^i's sorted in a non-decreasing order. We also define $T_n^{i,+} = \inf\{T_m^i : T_m^i > T_n^i\}$ and $T_n^{i,-} = \sup\{T_m^i : T_m^i < T_n^i\}$. Then we let $V_n^i = T_n^{i,+} - T_n^i$. We assume that V_n^i has CDF $F(\cdot)$ for $n \geq 1$ and $n \leq -2$. As we shall explain, the V_n^i's are *activities* which are executed by the i-th server in the upper bound process.

Next, we define, for each $i \in \{0, 1, ..., c\}$, and any $u \in (-\infty, \infty)$, the counting process

$$N_u^i(t) := \left|[u, u+t] \cap \mathcal{T}^i\right|,$$

for $t \geq 0$, where $|\ |$ denotes cardinality. Note that as $T_{-1}^i < 0 < T_1^i$ by stationary, $N_0^i(0) = 0$. For simplicity in the notation let us write $N^i(t) = N_0^i(t)$ if $t \geq 0$ and $N^i(t) = N_t^i(-t)$ if $t \leq 0$.

The quantity $N_u^0(t)$ is the number of customers who arrive during the time interval $[u, u+t]$. In the upper bound process, each of the c servers performs two types of activities: services and vacations. $N_u^i(t)$ is the number of activities initiated by server i during the time interval $[u, u+t]$.

2.1.1 The upper bound process

We shall refer to the upper bound process as the *vacation system*, for reasons which will become apparent. Let us explain first in words how does the vacation system operate. Customers arrive to the vacation system according to $\mathcal{T}^0$, and the system operates similarly to a $GI/GI/c$ queue, except that, every time a server (say server i^*) finishes an activity (i.e. service or a vacation), if there is no customer waiting to be served in the queue, server i^* takes a vacation which has the same distribution as the service time distribution; if there is at least one customer waiting, such customer starts to be served by server i^*. Similar vacation models have been used in [21] and [12].

More precisely, let $Q_v(t)$ denote the number of people waiting in queue at time t in the stationary vacation system. We write $Q_v(t_-) := \lim_{s\uparrow t} Q_v(s)$ and $dQ_v(t) := Q_v(t) - Q_v(t_-)$. Also, for for any $t \geq 0$, $i \in \{0, ..., c\}$ and each $u \in (-\infty, \infty)$, define

$$N_u^i(t_-) := \lim_{h\uparrow 0} N_{u-h}^i(t),$$

and let $dN_u^i(t) := N_u^i(t) - N_u^i(t_-)$ for all $t \geq 0$ (note that as $N_u^i(0_-) = 0$, $dN_u^i(0) = N_u^i(0_-)$). Similarly, for $t \leq 0$, $N^i(t_-) = N_t^i(|t|_-)$.

We also introduce $X_u(t) := N_u^0(t) - \sum_{i=1}^c N_u^i(t)$. Then the dynamics of $(Q_v(t) : t > 0)$ satisfy

$$dQ_v(t) = dX_0(t) + I(Q_v(t_-) = 0) \sum_{i=1}^{c} dN_0^i(t), \tag{1}$$

given $Q_v(0)$. Note that here we are using the fact that arrivals do not occur at the same time as the start of activity times; this is because the processes $\mathcal{T}^i$ are time stationary (and independent) renewal processes in continuous time so that T_{-1}^i and T_1^i have a density.

It follows from standard arguments for Skorokhod mapping [6] that for $t \geq 0$

$$Q_v(t) = Q_v(0) + X_0(t) - \inf_{0\leq s\leq t} \left((X_0(s) + Q_v(0))^- \right),$$

where $(X_0(s) + Q_v(0))^- = \min(X_0(s) + Q_v(0), 0)$. Moreover, using Lyons construction we have that $t \geq 0$

$$Q_v(-t) = \sup_{s\geq t} X_{-s}(0) - X_{-t}(0) \tag{2}$$

(see, for example, Proposition 1 of [3]). $(Q_v(t) : t \in (-\infty, \infty))$ is a well defined process by virtue of the stability condition $\lambda/(\mu c) < 1$.

The vacation system and the target process (the $GI/GI/c$ queue) will be coupled by using the same arrival stream of customers, $\mathcal{T}^0$, and assuming that each customer brings his own service time. In particular, the evolution of the underlying $GI/GI/c$ queue is described using a sequence of the form $\left((T_n^0, V_n) : n \in \mathbb{Z}\backslash\{0\} \right)$, where V_n is the service requirement of the customer arriving at time T_n^0. The V_n's must be extracted from the evolution of $Q_v(\cdot)$ so that the same service times are matched to the common arrival stream both in the vacation system and in the target process.

2.1.2 The coupling: extracting service times for each costumer

In order to match the service times corresponding to each of the arriving customers in the vacation system we define the following auxiliary processes. For every $i \in \{1, ..., c\}$, any $t > 0$, and each $u \in (-\infty, \infty)$, let $\sigma_u^i(t)$ denote the number of service initiations by server i during the time interval $[u, u+t]$. Observe that

$$\sigma_u^i(t) = \int_{[u,u+t]} I(Q_v(s_-) > 0)\, dN_u^i(s).$$

That is, we count service initiations which start at time $T_k^i \in [u, u+t]$ if and only if $Q_v\left(T_{k-}^i\right) > 0$. Once again, here we use that arrival times and activity initiation times do not occur simultaneously.

We now explain how to match the service time of the customer arriving at T_n^0. First, such customer occupies position $Q_v\left(T_n^0\right) \geq 1$ when he enters the queue. Let D_n^0 be the delay (or waiting time) inside the queue of the customer arriving at T_n^0, then we have that

$$D_n^0 = \inf\{t \geq 0 : Q_v\left(T_n^0\right) = \sum_{i=1}^{c} \sigma_{T_n^0}^i(t)\},$$

and therefore,

$$V_n = \sum_{i=1}^{c} V^i_{N^i(T^0_n + D^0_n)} \cdot dN^i\left(T^0_n + D^0_n\right). \tag{3}$$

Observe that the previous equation is valid because there is a unique $i\,(n) \in \{1, ..., c\}$ for which $dN^{i(n)}\left(T^0_n + D^0_n\right) = 1$ and $dN^j\left(T^0_n + D^0_n\right) = 0$ if $j \neq i\,(n)$ (ties are not possible because of the time stationarity of the $\mathcal{T}^i$s), so we obtain that (3) is equivalent to

$$V_n = V^{i(n)}_{N^{i(n)}(T^0_n + D^0_n)}.$$

We shall explain in the appendix to this section, that $(V_n : n \in \mathbb{Z}\backslash\{0\})$ and $\left(T^0_n : n \in \mathbb{Z}\backslash\{0\}\right)$ are two independent sequences and the V_n's are iid copies of V.

2.2 A family of $GI/GI/c$ queues and the target $GI/GI/c$ stationary system

We now describe the evolution of a family of standard $GI/GI/c$ queues. Once we have the sequence $\left(\left(T^0_n, V_n\right) : n \in \mathbb{Z}\backslash\{0\}\right)$ we can proceed to construct a family of continuous-time Markov processes $(Z_u(t; z) : t \geq 0)$ for each $u \in (-\infty, \infty)$, given the initial condition $Z_u\,(0; z) = z$. We write $z = (q, r, e)$, and set

$$Z_u(t; z) := \left(Q_u\,(t; z)\,, R_u\,(t; z)\,, E_u\,(t; z)\right),$$

for $t \geq 0$, where $Q_u\,(t; z)$ is the number of people in the queue at time $u + t$ $(Q_u\,(0; z) = q)$, $R_u(t; z)$ is the vector of ordered (ascending) remaining service times of the c servers at $u + t$ $(R_u(0; z) = r)$, and $E_u(t; z)$ is the time elapsed since the previous arrival at $u + t$ $(E_u(0; z) = e)$.

We shall always use $E_u(0; z) = e = u - \sup\{T^0_n : T^0_n \leq u\}$ and we shall select q and r appropriately based on the upper bound. The evolution of the process $(Z_u\,(s; z) : 0 < s \leq t)$ is obtained by feeding the traffic $\{\left(T^0_n, V_n\right) : u < T^0_n \leq u + s\}$ for $s \in (0, t]$ into a FCFS $GI/GI/c$ queue with initial conditions given by z. Constructing $(Z_u\,(s; z) : 0 < s \leq t)$ using the traffic trace $\{\left(T^0_n, V_n\right) : u < T^0_n \leq u + s\}$ for $s \in (0, t]$ is standard (see for example Chapter 3 of [18]).

One can further describe the evolution of the underlying $GI/GI/c$ at arrival epochs, using the Kiefer-Wolfowitz vector [1]. In particular, for every non-negative vector $w \in \mathbb{R}^c$, such that $w^{(i)} \leq w^{(i+1)}$ (where $w^{(i)}$ is the i-th entry of w), and each $k \in \mathbb{Z}\backslash\{0\}$ the family of processes $\{W_k\left(T^0_n; w\right) : n \geq k, n \in \mathbb{Z}\backslash\{0\}\}$ satisfies

$$\begin{aligned} W_k\left(T^{0,+}_n; w\right) &= \mathcal{S}\left(\left(W_k\left(T^0_n; w\right) + V_n \mathbf{e}_1 - A_n \mathbf{1}\right)^+\right), \\ W_k\left(T^0_k; w\right) &= w. \end{aligned} \tag{4}$$

where $\mathbf{e}_1 = (0, 0, ..., 1)^T \in \mathbb{R}^c$, $\mathbf{1} = (1, ..., 1)^T \in \mathbb{R}^c$, and $\mathcal{S}$ is the sorting operator which arranges the entries in a vector in ascending order. In simple words, $W_k\left(T^0_n; w\right)$ for $k \geq 1$ describes the Kiefer-Wolfowitz vector as observed by the customer arriving at T^0_n, assuming that customer who arrived at T^0_k, $k \leq n$, experienced the Kiefer-Wolfowitz state w.

Recall that the first entry of $W_k\left(T^0_n, w\right)$, namely $W^{(1)}_k\left(T^0_n, w\right)$, is the waiting time of the customer arriving at T^0_n (given the initial condition w at T^0_k). More generally, the i-th entry of $W_k\left(T^0_n; w\right)$, namely, $W^{(i)}_k\left(T^0_n; w\right)$, is the virtual waiting time of the customer arriving at T^0_n if he decided to enter service immediately after there are at least i servers free once he reaches the head of the line. In other words, one can also interpret $W_k\left(T^0_n; w\right)$ as the remaining vector of workloads (sorted in ascending order) that would be processed by each of the c servers at T^0_n, if no more arrivals got into the system after time T^0_n.

We are now ready to construct the stationary version of the $GI/GI/c$ queue. Namely, for each $n \in \mathbb{Z}\backslash\{0\}$ and every $t \in (-\infty, \infty)$ we define $W\,(n)$ and $Z\,(t)$ via

$$\begin{aligned} W\,(n) &:= \lim_{k \to -\infty} W_k\left(T^0_n; 0\right), \\ Z\,(t) &:= \left(Q\,(t)\,, R\,(t)\,, E\,(t)\right) = \lim_{u \to -\infty} Z_u\,(t - u, z_-)\,, \end{aligned} \tag{5}$$

where $z_- = (0, 0, e)$.

We shall show in Proposition 1 that these limits are well defined.

2.3 The analogue of the Kiefer-Wolfowitz process for the upper bound system

In order to complete the coupling strategy we also describe the evolution of the analog Kiefer-Wolfowitz vector induced by the vacation system, which we denote by $\left(W_v\left(T_n^0\right) : n \in \mathbb{Z}\backslash\{0\}\right)$, where v stands for vacation. As with the i-th entry of the Kiefer-Wolfowitz vector of a $GI/GI/c$ queue, the i-th entry of $W_v\left(T_n^0\right)$, namely $W_v^{(i)}\left(T_n^0\right)$, is the virtual waiting time of the customer arriving at time T_n^0 if he decided to enter service immediately after there are at least i servers free once he reaches the head of the line (assuming that servers become idle once they see, after the completion of a current activity, the customer in question waiting in the head of the line).

To describe the Kiefer-Wolfowitz vector induced by the vacation system precisely, let $U^i(t)$ be the time until the next renewal after time t in $\mathcal{T}^i$, that is $U^i(t) = \inf\{T_n^i : T_n^i > t\} - t$. So, for example, $U^0\left(T_n^0\right) = A_n$ for $n \geq 1$. Let $U(t) = \left(U^1(t), ..., U^c(t)\right)^T$. We then have that

$$W_v\left(T_n^0\right) = D_n^0\mathbf{1} + \mathcal{S}\left(U\left(\left(T_n^0 + D_n^0\right)_-\right)\right). \tag{6}$$

In particular, note that $W_v^{(1)}\left(T_n^0\right) = D_n^0$.

Actually, in order to draw a closer connection to the Kiefer-Wolfowitz vector recursion of a standard $GI/GI/c$ queue, let us write

$$\mathcal{S}\left(U\left(\left(T_n^0 + D_n^0\right)_-\right)\right) = (U^{(1)}\left(\left(T_n^0 + D_n^0\right)_-\right), ..., U^{(c)}\left(\left(T_n^0 + D_n^0\right)_-\right)^T,$$

and suppose that $U^{(i)}\left(\left(T_n^0 + D_n^0\right)_-\right) = U^{j_i(n)}\left(\left(T_n^0 + D_n^0\right)_-\right)$ (i.e. $j_i(n)$ is the server whose remaining activity time right before $T_n^0 + D_n^0$ is the i-th smallest in order). Then, define

$$\bar{W}_v\left(T_n^0\right) = W_v\left(T_n^0\right) + V_n\mathbf{e}_1 - A_n\mathbf{1},$$

and let $\bar{W}_v^{(i)}\left(T_n^0\right)$ to be the i-th entry of $\bar{W}_v\left(T_n^0\right)$. It is not difficult to see from the definition of $W_v\left(T_n^0\right)$ that

$$W_v^{(i)}\left(T_n^{0,+}\right) = \mathcal{S}\left(\left(\bar{W}_v^{(i)}\left(T_n^0\right)\right)^+ + \Xi_n^{(i)}\right),$$

where

$$\begin{aligned} \Xi_n^{(i)} &= I(\bar{W}_v^{(i)}\left(T_n^0\right) < 0) \cdot U^{j_i(n)}\left(\left(T_n^{0,+} + D_n^{0,+}\right)_-\right), \text{ and} \\ D_n^{0,+} &= \inf\{D_k^0 : D_k^0 > D_n^0\}. \end{aligned}$$

So, (6) actually satisfies

$$W_v\left(T_n^{0,+}\right) = \mathcal{S}\left(\left(W_v\left(T_n^0\right) + V_n\mathbf{e}_1 - A_n\mathbf{1}\right)^+ + \Xi_n\right), \tag{7}$$

where $\Xi_n = \left(\Xi_n^{(1)}, ..., \Xi_n^{(c)}\right)$.

2.4 Description of simulation strategy and main result

We now describe how the variation of DCFTP that we described in the Introduction, using the monotonicity of the multiserver queue, and elements a) to d) apply to our setting.

The upper bound is initialized using the Kiefer-Wolfowitz process associated to the vacation system. For the lower bound, we shall simply pick the null vector.

The strategy combines the following facts (which we shall discuss in the sequel).

- **Fact I:** We can simulate $\sup_{s\geq t} X_{-s}(0)$, $\left(N_{-t}^i(0) : 1 \leq i \leq c\right)$, $\left(N_0^i(t) : 1 \leq i \leq c\right)$, jointly for any given $t \geq 0$. This part, which corresponds to item c), is executed using an algorithm from [5] designed to sample the infinite horizon running time maximum of a random walk with negative drift. We shall explain in Section 5 how the algorithm in [5] can be easily modified to sample $\sup_{s\geq t} X_{-s}(0)$ jointly with $(N_{-s}^i(0) : s \geq 0)_{i=0}^c$.

- **Fact II:** For all $k \leq -1$ and every $k \leq n \leq -1$ we have that

$$W_k\left(T_n^0; 0\right) \leq W\left(n\right) \leq W_k\left(T_n^0; W_v\left(T_k^0\right)\right).$$

  This portion exploits the upper bound a) (i.e. $W_v\left(T_k^0\right)$), and the lower bound b) (i.e. 0).

- **Fact III:** We can detect that coalescence occurs at some time $-T \in [T_\kappa^0, 0]$ for some $\kappa \leq -1$ by finding $n \in \mathbb{Z}_-$, $n \geq \kappa$, such that $T_n^0 + W_\kappa^{(1)}\left(T_n^0; W_v\left(T_\kappa^0\right)\right) \leq 0$ and

$$W_\kappa\left(T_n^0; W_v\left(T_\kappa^0\right)\right) = W_\kappa\left(T_n^0; 0\right).$$

  This portion is precisely the coalescence detection strategy which uses monotonicity of the Kiefer-Wolfowitz vector.

- **Fact IV**: We can combine Facts I-III to conclude that

$$Z_{T_\kappa^0}\left(\left|T_\kappa^0\right|; Q\left(T_{-\kappa}^0\right), \mathcal{S}\left(U\left(T_{-\kappa}^0\right)\right), 0\right) = Z\left(0\right) \tag{8}$$

  is stationary. And we also have that

$$W_\kappa\left(T_1^0; 0\right) = W\left(1\right)$$

  follows the stationary distribution of the Kiefer-Wolfowitz vector of a $GI/GI/c$ queue.

The main result of this paper is the following.

Theorem 1. *Assume (A1) is in force, with $\lambda/(\mu c) \in (0,1)$. Then Facts I-IV hold true and (8) is a stationary sample of the state of the multi-server system. we can detect coalescence at a time $-T < 0$ such that $E\left(T\right) < \infty$.*

The rest of the paper is dedicated to the proof of Theorem 1. In Section 3 we verify a number of monotonicity properties which in particular allows us to conclude that the construction of $W\left(n\right)$ and $Z\left(t\right)$ is legitimate (i.e. that the limits exist almost surely). This monotonicity properties also yield Fact II and pave the way to verify Fact III. Section 4 proves the finite expectation of the coalescence time. In Section 5 we give more details as how to carry out the simulation of the upper bound process. But before going over Facts, we conclude this section, as we promised, arguing that the V_n's are iid and independent of the arrival sequence $\mathcal{T}^0$.

2.5 Appendix: The iid property of the coupled service times and independence of the arrival process

In order to explain why the V_n's form an iid sequence, independent of the sequence $\mathcal{T}^0 = \{T_n^0 : n \in \mathbb{Z}\backslash\{0\}\}$, it is useful to keep in mind the diagram depicted in Figure 1, which illustrates a case involving two servers, $c = 2$.

The assignment of the service times, as we shall explain, can be thought of as a procedure similar to a tetris game. Arrival times are depicted by dotted horizontal lines which go from left to right starting at the left most vertical line, which is labeled "Arrivals". Think of the time line going, vertically, from the bottom of the graph (past) to the top of the graph (future).

In the right most column in Figure 1 we indicate the queue length, right at the time of a depicted arrival (and thus, including the arrival itself). So, for example, the first arrival depicted in Figure 1 observes one customer waiting and thus, including the arrival himself, there are two customers waiting in queue.

The tetris configuration observed by an arrival at time T is comprised of two parts: i) the receding horizon, which corresponds to the remaining incomplete blocks, and ii) the landscape, comprised of the configuration of complete blocks. So, for example, the first arrival in Figure 1 observes a receding horizon corresponding to the two white remaining blocks which start from the dotted line at the bottom. The landscape can be parameterized by a sequence of block sizes and the order of the sequence is given by the way in which the complete blocks appear from bottom to top – this is precisely the tetris-game assignment. There are no

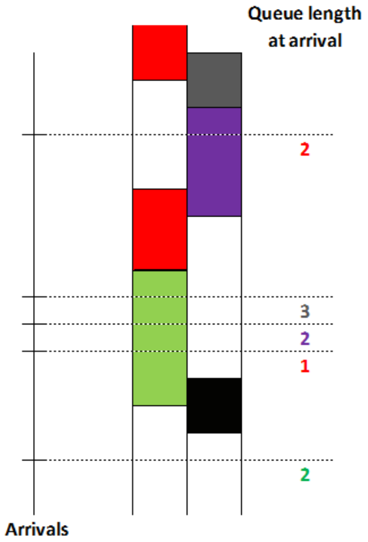


Figure 1: Matching Procedure of Service Times to Arrival Process

ties because of the continuous time stationarity and independence of the underlying renewal processes. The colors are, for the moment, not part of the landscape. We will explain the meaning of the colors momentarily.

The assignment of the service times is done as follows. The arriving customer reads off the right-most column (with heading "Queue length at arrival") and selects the block size labeled precisely with the number indicated by the "Queue length at arrival". So, there are two distinctive quantities to keep in mind assigned to each player (i.e. arriving customer): a) the landscape (or landscape sequence, which, as indicated can be used to reconstruct the landscape), and b) the *service time*, which is the complete block size occupying the "Queue length at arrival"-th position in the landscape sequence.

The color code in Figure 1 simply illustrates quantity b) for each of the arrivals. So, for example, the first arrival, who reads "Queue length at arrival = 2" (which we have written in green color), gets assigned the second complete block, which we have depicted in green. Similarly, the second arrival depicted, reads off the number "1" (written in red) and gets assigned the first red block depicted (from bottom to top). The very first complete block (from bottom to top), which is depicted in black, corresponds to the service time assigned to the customer ahead of the customer who collected the green block. The number "1" (in red) is obtained by observed that the customer with the initial black block has departed.

Now we argue the following properties:

1) The service times are iid copies of V.

2) The service times are independent of $\mathcal{T}^0$.

About property 1): The player arriving at time T, reads a number, corresponding to the queue length, which is obtained by the *past filtration* $\mathcal{F}_T$ generated by $\cup_{k\in\mathbb{Z}\setminus\{0\},0\leq i\leq c}\{T_k^i : T_k^i \leq T\}$. Conditional on the receding horizon (i.e. remaining incomplete block sizes), $\mathcal{R}_T$, the past filtration is independent of the landscape. This is simply the Markov property applied to the forward residual life time process of each of the c renewal processes represented by the c middle columns. Moreover, conditional on $\mathcal{R}_T$, each landscape forms a sequence of iid copies of V because of the structure of the underlying c renewal processes corresponding to the middle columns. So, let $Q(T)$ denote the queue length T (including the arrival at time T), which is a function of the past filtration, and let $\{L_T(k) : k \geq 1\}$ be the landscape sequence observed at time T, so that $L_T(Q(T))$ is the service time of the customer who arrives at time T. We then have that for any positive and bounded continuous function $f(\cdot)$

$$E[f(L_T(Q(T)))\,|\mathcal{R}_T] = E[f(L_T(1))\,|\mathcal{R}_T] = E[f(V)\,|\mathcal{R}_T],$$

precisely because conditional on $\mathcal{R}_T$, $Q(T)$ (being $\mathcal{F}_T$ measurable) is independent of L_T.

To verify the iid property, let f_1, f_2 be non-negative and bounded continuous functions. And assume that $T_1 < T_2$ are arrival times in $\mathcal{T}^0$ (not necessarily consecutive). Then,

$$\begin{aligned}
&E[f_1\left(L_{T_1}\left(Q\left(T_1\right)\right)\right) f_2\left(L_{T_2}\left(Q\left(T_2\right)\right)\right)] \\
&= E[E[f_1\left(L_{T_1}\left(Q\left(T_1\right)\right)\right) f_2(L_{T_2}\left(Q\left(T_2\right)\right))|\mathcal{F}_{T_2}, \mathcal{R}_{T_2}]] \\
&= E[f_1\left(L_{T_1}\left(Q\left(T_1\right)\right)\right) E[f_2(L_{T_2}\left(Q\left(T_2\right)\right))|\mathcal{F}_{T_2}, \mathcal{R}_{T_2}]] \\
&= E[f_1(L_{T_1}\left(Q\left(T_1\right)\right))]E[f_2(V)] = E[f_1(V)]E[f_2(V)].
\end{aligned}$$

The same argument extends to any subset of arrival times and thus the iid property follows.

About property 2): Note that in the calculations involving property 1), the actual values of the arrival times T, and T_1 and T_2 are irrelevant. The iid property of the service times is established path-by-path conditional on the observed realization $\mathcal{T}^0$. Thus the independence of the arrival process and service times follows immediately.

3 Monotonicity Properties and Stationary $GI/GI/c$ System

This section we will present several lemmas which contain useful monotonicity properties. The proofs of the lemmas are given at the end of this section in order to quickly arrive to the main point of this section, which is the construction of a stationary version of the $GI/GI/c$ queue.

First we recall that the Kiefer-Wolfowitz vector of $GI/GI/c$ queue is monotone in the initial condition (9) and invoke a property (10) which will allows us to construct a stationary version of the Kiefer-Wolfowitz vector of our underlying $GI/GI/c$ queue, using Lyons construction.

Lemma 1. *For $n \geq k$, $k, n \in \mathbb{Z}\backslash\{0\}$, $w^+ > w^-$,*

$$W_k\left(T_n^0; w^+\right) \geq W_k\left(T_n^0; w^-\right). \tag{9}$$

Moreover, if $k \leq k' \leq n$

$$W_k\left(T_n^0; 0\right) \geq W_{k'}\left(T_n^0; 0\right). \tag{10}$$

The second result allows to make precise a sense in which the vacation system dominates a suitable family of $GI/GI/c$ systems, in terms of the underlying Kiefer-Wolfowitz vectors.

Lemma 2. *For $n \geq k$, $k, n \in \mathbb{Z}\backslash\{0\}$,*

$$W_v\left(T_n^0\right) \geq W_k\left(T_n^0; W_v\left(T_k^0\right)\right).$$

The next result shows that in terms of queue length processes, the vacation system also dominates a family of $GI/GI/c$ queue, which we shall use to construct the upper bounds.

Lemma 3. *Let $q = Q_v(u)$, $r = \mathcal{S}\left(U\left(u\right)\right)$, and $e = u - \sup\{T_n^0 : T_n^0 \leq u\}$, so that $z^+ = (q, r, e)$ and $z^- = (0, 0, e)$ then for $t \geq u$*

$$Q_u(t-u; z^-) \leq Q_u(t-u; z^+) \leq Q_v(t).$$

Using Lemmas 1, 2, and 3 we can establish the following result.

Proposition 1. *The limits defining $W\left(n\right)$ and $Z\left(t\right)$ in (5) exist almost surely. Moreover, we have that Fact II holds.*

Proof of Proposition 1. Using Lemma 2 and Lemma 1 we have that

$$W_v\left(T_n^0\right) \geq W_k\left(T_n^0; W_v\left(T_k^0\right)\right) \geq W_k\left(T_n^0; 0\right).$$

So, by property (10) in Lemma 1 we conclude that the limit defining $W\left(n\right)$ exists almost surely and that

$$W\left(n\right) \leq W_v\left(T_n^0\right). \tag{11}$$

Similarly, using Lemma 3 we can obtain the existence of the limit $Q(t)$ and we have that $Q\left(t\right) \leq Q_v\left(t\right)$. Moreover, by convergence of the Kiefer-Wolfowitz vectors we obtain the i-th entry of $R\left(T_n^0 + W^{(1)}(n)\right)$, namely, $R^{(i)}\left(T_n^0 + W^{(1)}(n)\right) = \left(W^{(i)}\left(n\right) - W^{(1)}\left(n\right)\right)^+$, where $i \in \{1, ..., c\}$. Clearly, since the age process has been taken underlying $\mathcal{T}^0$, we have that $E\left(t\right) = t - \sup\{T_n^0 : T_n^0 \leq t\}$. The fact that the limits are stationary follows directly from the limiting procedure and it is standard in Lyons-type constructions. For Fact II, we use the identity $W\left(n\right) = W_k\left(T_n^0; W\left(k\right)\right)$, combined with Lemma 1 to obtain

$$W_k\left(T_n^0; 0\right) \leq W_k\left(T_n^0; W\left(k\right)\right) = W\left(n\right),$$

and then we apply Lemma 2, together with (11), to obtain

$$W\left(n\right) = W_k\left(T_n^0; W\left(k\right)\right) \leq W_k\left(T_n^0; W_v\left(T_k^0\right)\right).$$

The previous two inequalities are precisely the statement of Fact II. □

3.1 Proof of technical Lemmas

Proof of Lemma 1. Both facts are standard, the first one can be easily shown using induction. Specifically, we first notice that $W_k(T_k^0; w^+) = w^+ > w^- = W_k(T_n^0; w^-)$ Suppose that $W_k(T_n^0; w^+) \geq W_k(T_n^0; w^-)$ for some $n \geq 0$, then

$$\begin{aligned} W_k(T_{n+1}^0; w^+) &= \mathcal{S}\left(\left(W_k(T_n^0; w^+) + V_n - A_n\right)^+\right) \\ &\geq \mathcal{S}\left(\left(W_k(T_n^0; w^-) + V_n - A_n\right)^+\right) = W_k(T_{n+1}^0; w^-). \end{aligned}$$

For inequality (10), we note that $W_k\left(T_{k'}^0; 0\right) \geq W_{k'}\left(T_{k'}^0; 0\right) = 0$, and therefore, due to (9), we have that

$$W_k\left(T_n^0; 0\right) = W_{k'}\left(T_n^0; W_k\left(k'; 0\right)\right) \geq W_{k'}\left(T_n^0; 0\right).$$

□

Proof of Lemma 2. This fact follows immediately by induction from equations (4) and (7) using the fact that $\Xi_n \geq 0$. □

Proor of Lemma 3. We first prove the inequality $Q_u(t-u; z^+) \leq Q_v(t)$. Note that $U^i\left(u\right) > 0$ for all u (the forward residual life time process is right continuous), so the initial condition r indicates that all the servers are busy (operating) and the initial $q \geq 0$ customers will leave the queue (i.e. enter service) at the same time in the vacation system as under the evolution of $Z_u\left(\cdot; z^+\right)$. Now, let us write $N = \inf\{n : T_n^0 \geq u\}$ (in words, the next arriving customer at or after u arrives at time T_N^0). It is easy to see that $\mathcal{S}\left(U\left(T_N^0\right)\right) \geq R_u(T_N^0 - u; z^+)$; to wit, if T_N^0 occurs before any of the servers becomes idle, then we have equality, and if T_N^0 occurs after, say $l \geq 1$, servers become idle, then $R_u(T_N^0 - u; z^+)$ will have l zeroes and the bottom $c - l$ entries will coincide with those of $\mathcal{S}\left(U\left(T_N^0\right)\right)$, which has strictly positive entries. So, if w_N is the Kiefer-Wolfowitz vector observed by the customer arriving at T_N^0 (induced by $Q_u(\cdot - u; z^+)$), then we have $W_v\left(T_N^0\right) \geq w_N$. By monotonicity of the Kiefer-Wolfowitz vector in the initial condition and because of Lemma 2, we have

$$W_v\left(T_k^0\right) \geq W_N\left(T_k^0, W_v\left(T_N^0\right)\right) \geq W_N\left(T_k^0, w_N\right),$$

for all $k \geq N$, and hence, $T_k^0 + D_k^0 \geq T_k^0 + W_N^{(1)}\left(T_k^0, w_N\right)$. Therefore, the departure time from the queue (i.e. initiation of service) of the customer arriving at T_k^0 in the vacation system occurs after the departure time from the queue of the customer arriving at time T_k^0 in the $GI/GI/c$ queue. Consequently, we conclude that the set of customers waiting in the queue in the $GI/GI/c$ system at time t is a subset of the set of customers waiting in the queue in the vacation system at the same time. Similarly, we consider $Q_u(t-u;z^-) \le Q_u(t-u;z^+)$, which is easier to establish, since for $k \ge N$ (with the earlier definition of T_N^0 and w_N),

$$W_N\left(T_k^0;w_N\right) \ge W_N\left(T_k^0;0\right),$$

So the set of customers waiting in the queue in the lower bound $GI/GI/c$ system at time t is a subset of the set of customers waiting in the upper bound $GI/GI/c$ system at the same time. □

4 The coalescence detection in finite time

In this section, we give more details about the coalescence detection scheme. The next result corresponds to Fact III and Fact IV.

Proposition 2. *Suppose that* $w^+ = W_v\left(T_k^0\right)$ *and* $w^- = 0$. *Assume that* $W_k\left(T_n^0;w^+\right) = W_k\left(T_n^0;w^-\right)$ *for some* $k \le n \le -1$. *Then,* $W_k\left(T_m^0;w^+\right) = W\left(m\right) = W_k\left(T_m^0;w^-\right)$ *for all* $m \ge n$. *Moreover, for all* $t \ge T_n^0 + W_k^{(1)}\left(T_n^0;w^+\right)$,

$$Z_{T_k^0}(t-T_k^0;Q_v\left(T_k^0\right),\mathcal{S}\left(U\left(T_k^0\right)\right),0) = Z_{T_k^0}(t-T_k^0;0,0,0) = Z\left(t\right). \tag{12}$$

Proof of Proposition 2. The fact that

$$W_k\left(T_m^0;w^+\right) = W\left(m\right) = W_k\left(T_m^0;w^-\right)$$

for $m \ge n$ follows immediately from the recursion defining the Kiefer-Wolfowitz vector. Now, to show the first equality in (12) it suffices to consider $t = T_n^0 + W_k^{(1)}\left(T_n^0;w^+\right)$, since from $t \ge T_n^0$ the input is exactly the same and everyone coming after T_n^0 will depart the queue and enter service after $T_n^0 + W_k^{(1)}\left(T_n^0;w^+\right)$. The arrival processes (i.e. $E_u\left(\cdot\right)$) clearly agree, so we just need to verify that the queue lengths and the residual service times agree. First, note that

$$\begin{aligned}
&R_{T_k^0}(T_n^0 + W_k^{(1)}\left(T_n^0;w^+\right) - T_k^0;Q_v\left(T_k^0\right),\mathcal{S}\left(U\left(T_k^0\right)\right),0)\\
&= W_k\left(T_n^0;w^+\right) - W_k^{(1)}\left(T_n^0;w^+\right)\mathbf{1}\\
&= W_k\left(T_n^0;w^-\right) - W_k^{(1)}\left(T_n^0;w^-\right)\mathbf{1}\\
&= R_{T_k^0}(T_n^0 + W_k^{(1)}\left(T_n^0;w^-\right) - T_k^0;0,0,0).
\end{aligned} \tag{13}$$

So, the residual service times of both upper and lower bound processes agree. The agreement of the queue lengths follows from Lemma 3. Finally, the second equality in (12) is obtained by taking the limit in the last equality in (13), and the equality between queue lengths follows again from Lemma 3. □

Next we analyze properties of the coalescence time. Define

$$T_- = \sup\{T_k^0 \le 0 : \inf_{T_k^0 \le t \le 0}[Z_{T_k^0}(t-T_k^0;Q_v\left(T_k^0\right),\mathcal{S}\left(U\left(T_k^0\right)\right),0) - Z_{T_k^0}(t-T_k^0;0,0,0)] = 0\}.$$

By time reversibility we have that $|T_-|$ is equal in distribution to

$$T = \inf\{T_k^0 \ge 0 : \inf_{0 \le t \le T_k^0}[Z_0(t;Q_v\left(0\right),\mathcal{S}\left(U\left(0\right)\right),0) - Z_0(t;0,0,0)] = 0\}.$$

We next establish that $E[T] < \infty$. This result, except for the verification of Fact I, which will be discussed in Section 5, completes the proof of Theorem 1.

Proposition 3. *If* $E[V_n] < cE[A_n]$ *for* $n \ge 1$ *and Assumption (A1) holds,*

$$E[T] < \infty.$$

Proof of Proposition 3. Define

$$\tau = \inf\left\{n \geq 1 : W_1\left(T_n^0; W_v\left(1\right)\right) = W_1\left(T_n^0; 0\right)\right\}.$$

By Wald's identity, $EA_n < \infty$, for any $n \geq 1$, it suffices to show that $E[\tau] < \infty$. We prove the proposition by considering a sequence of events which happens with positive probability and leads to the occurrence of τ. The idea follows from the proof of Lemma 2.4 in Chapter XII of [1]. As $E[V_n] < cE[A_n]$, for $n \geq 2$,we can find $m, \epsilon > 0$ such that for every $n \geq 2$, the event $H_n = \{V_n < cm - \epsilon, A_n > m\}$ is nontrivial in the sense that $P\left(H_n\right) > \delta$ for some $\delta > 0$. Moreover, because V_1 and A_1 are independent with continuous distribution we also have that $P\left(H_1\right) > \delta$. Now, pick $K > cm$, large enough, and define

$$\Omega = \left\{W_1^{(c)}\left(T_k^0; W_v\left(1\right)\right) \leq K\right\} \bigcap_{n=k}^{k+\lceil cK/\epsilon\rceil+c} H_n.$$

In what follows, we first show that if Ω happens, the two bounding systems would have coalesced by the time of the $(k + c + \lceil cK/\epsilon\rceil)$-th arrival. We then provide an upper bound for $E[\tau]$. Let

$$\tilde{W}_k = W_1\left(T_k^0; W_v\left(1\right)\right).$$

For $n \geq k$, define $\tilde{V}_n = cm - \epsilon$, $\tilde{A}_n = m$. and the (auxiliary) Kiefer-Wolfowitz sequence

$$\tilde{W}_{n+1} = \mathcal{S}\left(\left(\tilde{W}_n + \tilde{V}_n\mathbf{e}_1 - \tilde{A}_n\mathbf{1}\right)^+\right).$$

Then Ω implies $V_n \leq \tilde{V}_n$ and $A_n > \tilde{A}_n$, for $n \geq k$, which in turn implies $W_1\left(T_n^0; W_v\left(1\right)\right) \leq \tilde{W}_n$. Moreover, It is easy to check that $\tilde{W}_n^{(1)} = 0$ and $\tilde{W}_n^{(c)} < cm$ for $n = k + \lceil cK/\epsilon\rceil + 1, \cdots, k + \lceil cK/\epsilon\rceil + c$. Then, $W_1^{(1)}\left(T_n^0; W_v\left(1\right)\right) = 0$ and $W_1^{(c)}\left(T_n^0; W_v\left(1\right)\right) < cm$ for $n = k+\lceil cK/\epsilon\rceil+1, \cdots, k+\lceil cK/\epsilon\rceil+c$. This indicates that under Ω, all the arrivals between the $(k + \lceil cK/\epsilon\rceil + 1)$-th arrival and the $(k + \lceil cK/\epsilon\rceil + c)$-th arrival (included) have zero waiting time (enter service immediately upon arrival), and the customers initially seen by the $(k + \lceil cK/\epsilon\rceil + 1)$-th arrival would have had left the system by the time of the $(k + \lceil cK/\epsilon\rceil + c)$-th arrival. The same analysis clearly holds assuming that we replace $W_1\left(T_k^0; W_v\left(1\right)\right)$ by $W_1\left(T_k^0; 0\right)$ throughout the previous discussion. Therefore, by the time of the $(k + \lceil cK/\epsilon\rceil + c)$-th arrival, the two bounding systems would have exactly the same set of customers with exactly the same remaining service times, which is equal to their service times minus the time elapsed since their arrival times (since all of them start service immediately upon arrival). We also notice that since there is no customer waiting, the sorted remaining service time at $T_{k+\lceil cK/\epsilon\rceil+c}^0$ coincide with the Kiefer-Wolfowitz vector $\tilde{W}_{k+\lceil cK/\epsilon\rceil+c}$. Under Assumption (A1) and $\lambda < c\mu$, $\{W\left(T_n^0; w\left(1\right)\right) : n \geq 1\}$ for any fixed initial condition w, is a positive recurrent Harris chain, [1] – in fact, positive recurrence follows even under the assumption of finite means. Now, again under Assumption (A1), we have that

$$E\left[\sum_{i=1}^{c} W_v^{(i)}\left(1\right)\right] < \infty,$$

Therefore, it is straightforward to show, using a standard linear Lyapunov function, that for K large enough, if $\tilde{\kappa}_1 := \inf\{n \geq 1 : W^{(c)}\left(T_n^0; W_v\left(1\right)\right) \leq K\}$, then $E[\tilde{\kappa}_1] < \infty$. Moreover, for $i \geq 2$, define

$$\tilde{\kappa}_i := \{n > \tilde{\kappa}_{i-1} + \lceil cK/\epsilon\rceil + c : W^{(c)}\left(n; W_v\left(1\right)\right) \leq K\}.$$

By the positive recurrence of the Kiefer-Wolfowitz vector, we can find a constant $M > 0$ such that

$$E[\tilde{\kappa}_i - \tilde{\kappa}_{i-1}] < M.$$

In this way, we can split the process into cycles (not necessarily regenerative) $[\tilde{\kappa}_i, \tilde{\kappa}_{i+1})$ for $i \geq 1$, and the initial period $[1, \tilde{\kappa}_1)$. We denote $\Omega_i = \bigcap_{n=\tilde{\kappa}_i}^{\tilde{\kappa}_i+\lceil cK/\epsilon\rceil+c} H_n$ for $i = 1, 2, \cdots$. Since $P(H_n) > \delta$, $P(\Omega_i) \geq \delta^{\lceil cK/\epsilon\rceil+c+1} > 0$.

Let $N = \inf\{i \geq 1 : I(\Omega_i = 1)\}$ (i.e. the first i for which Ω_i occurs), then $E[N] \leq \delta^{-(\lceil cK/\epsilon \rceil + c + 1)} < \infty$. By Wald's identity we have (setting $\tilde{\kappa}_0 = 0$) that

$$\begin{aligned} E[\tau] &\leq E[\tilde{\kappa}_N] + \lceil cK/\epsilon \rceil + c \\ &= E\sum_{i=1}^{N}(\tilde{\kappa}_i - \tilde{\kappa}_{i-1}) + \lceil cK/\epsilon \rceil + c \\ &\leq E[N] \times M + E[\tilde{\kappa}_1] + \lceil cK/\epsilon \rceil + c. \end{aligned}$$

□

5 Fact I: Simulation of Stationary Vacation System Backwards in Time

In this section, we address the validity of Fact I, namely, that we can simulate the vacation system backwards in time, jointly with $\left\{T_n^i : m \leq n \leq -1\right\}$ for $1 \leq i \leq c$ for any $m \leq -1$.

Let $G_e(\cdot) = \lambda \int_0^{\cdot} \bar{G}(x)dx$ and $F_e(\cdot) = \mu \int_0^{\cdot} \bar{F}(x)dx$ denote equilibrium CDF's of the interarrival time and service time distributions respectively. We first notice that simulating the stationary arrival process $\left\{T_n^0 : n \leq -1\right\}$ and stationary service/vacation completion process $\left\{T_n^i : n \leq -1\right\}$ for each $1 \leq i \leq c$ is straightforward by the reversibility of $\mathcal{T}_n^i$ for $0 \leq i \leq c$. Specifically, we can simulate the renewal arrival process forward in time from time 0 with the first interarrival time following G_e and subsequence interarrival times following G. We then set $T_{-k}^0 = -T_k^0$ for all $k \geq 1$. Likewise, we can also simulate the service/vacation completion process of server i, for $i = 1, \ldots, c$, forward in time from time 0 with the first service/vacation completion time following F_e and subsequent service/vacation requirements distributed as F. Let T_k^i denote the k-th service/vacation completion time of server i counting forwards (backwards) in time. Then we set $T_{-k}^i = -T_k^i$.

Similarly, we have the equality in distribution, for all $t \geq 0$ (jointly)

$$X_{-t}(0) = X_0(t),$$

therefore we have from (2) that the following equality in distribution holds for all $t \geq 0$ (jointly)

$$Q_v(-t) = \sup_{s \geq t} X_0(s) - X_0(t).$$

The challenge in simulating $Q_v(-t)$, involves in sampling $M(t) = \max_{s \geq t}\{X_0(s)\}$ jointly with $N_0^i(t)$ for $1 \leq i \leq c$ during any time interval of the form $[0, T]$ for $T > 0$. If we can do that, then we can evaluate

$$X_0(t) = N_0^0(t) - \sum_{i=1}^{c} N_0^i(t),$$

and, therefore, $Q_v(-t) = M(t) - X(t)$.

In what follows we first introduce a trick to decompose the process of sampling $M(t)$ into that of sampling the maximum of $c+1$ independent negative drifted random walks.

Choose $a \in (\lambda, c\mu)$. Then

$$X(t) = (N_0^0(t) - at) + \sum_{i=1}^{c}\left(\frac{a}{c}t - N_0^i(t)\right).$$

We define $(c+1)$ negative drifted random walks as follows:

$$S_0^{(0)} = 0, \quad S_n^{(0)} = S_{n-1}^{(0)} + (-aA_n + 1) \text{ for } n = 1, 2, \ldots,$$

and for $i = 1, \ldots, c$,

$$S_0^{(i)} = \frac{a}{c} V_1^{(i)}, \quad S_n^{(i)} = S_{n-1}^{(i)} + \left(-1 + \frac{a}{c} V_{n+1}^{(i)}\right) \text{ for } n = 1, 2, \ldots.$$

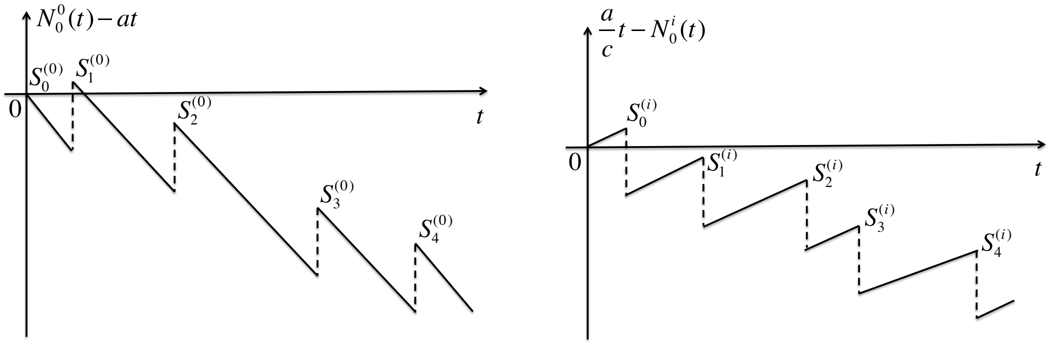
Figure 2 plots the relationship between $\{N_0^0(t) - at : t \geq 0\}$ and $\{S_n^{(0)} : n \geq 0\}$, and the relationship between $\{\frac{a}{c}t - N_0^i(t) : t \geq 0\}$ and $\{S_n^{(i)} : n \geq 0\}$ for $i = 1, \ldots, c$. From Figure 2, it is easy to check that

$$\max_{s \geq t} \{N_0^0(s) - as\} = \max\{N_0^0(t) - at, \max_{n \geq N_0^0(t)+1} \{S_n^{(0)}\}\},$$

and for $i = 1, \ldots, c$,

$$\max_{s \geq t} \left\{\frac{a}{c}s - N_0^i(s)\right\} = \max\{\frac{a}{c}t - N_0^i(t), \max_{n \geq N_0^i(t)} \{S_n^{(i)}\}\}.$$

Figure 2: The relationship between the renewal processes and the random walks



(a) $N_0^0(t) - at$ and $S_n^{(0)}$

(b) $(a/c)t - N_0^i(t)$ and $S_n^{(i)}$

Consequently, the algorithm to simulate $M(t) = \max_{s \geq t}\{X_0(s)\}$ jointly with $N_0^i(t)$ for $1 \leq i \leq c$ during any time interval of the form $[0, T]$ for $T > 0$ can be implemented if we can simulate $S_n^{(i)}$ jointly with $M_n^{(i)} = \max\{S_k^{(i)} : k \geq n\}$ for $n \geq 1$ for $i = 0, 1, \ldots, c$. The algorithm to generate $S_n^{(i)}$ jointly with $M_n^{(i)}$ under Assumption (A1) has been developed in [5]. We provide a detailed Matlab implementation of each of the algorithms required to execute Facts I-IV in the online appendix to this paper.

6 Numerical experiments

As a sanity check, we have implemented our Matlab code in the case of an $M/M/c$ queue, for which the steady-state analysis can be performed in closed form.

As a first step, we have compared the theoretical distribution to the empirical distribution obtained from a large number of runs of our perfect simulation algorithm for different sets of parameter values, and they are all in close agreement. As an example, Figure 3 shows the result of such test when $\lambda = 3$, $\mu = 2$, $c = 2$. Grey bars show the empirical result of $5,000$ draws using our perfect simulation algorithm, and black bars show the theoretical distribution of number of customers in system. Figure 4 provides another comparison with a different set of parameters.

Next we run numerical experiments to see how the running time of our algorithm, measured by mean coalescence time of two bounding systems, scales as the number of servers grows and the traffic intensity ρ changes. Starting from time 0, the upper bound queue has its queue length sampled from the theoretical distribution of an $M/M/c$ vacation system and all servers busy with remaining service times drew from the equilibrium distribution of the service/vacation time; and the lower bound queue is empty. Then we run

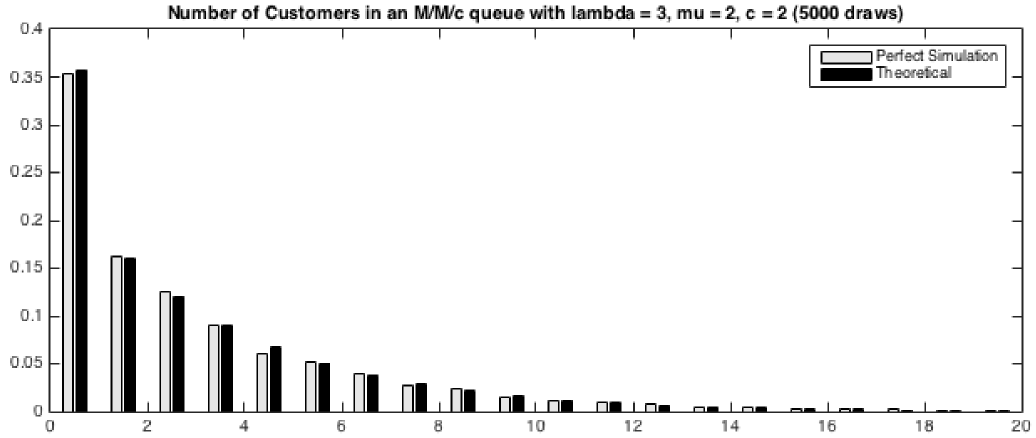


Figure 3: Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 3$, $\mu = 2$ and $c = 2$.

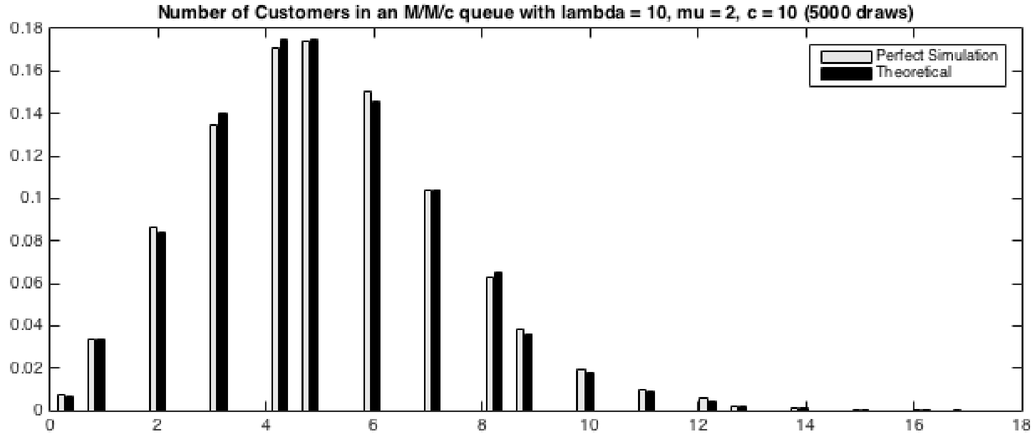


Figure 4: Number of customers for an $M/M/c$ queue in stationarity when $\lambda = 10$, $\mu = 2$ and $c = 10$.

both the upper bound and lower bound queues forward in time with the same stream of arrival times and service requirements until they coalescence. Table 1 shows the estimated average coalescence time, $E[T]$, based on 5000 iid samples, for different system scales in the Quality driven regime (QD). We observe that $E[T]$ does not increase much as the system scale parameter, s, grows. Table 2 shows similar results for the Quality-and-Efficiency driven operating regime (QED). In this case, $E[T]$ increases at a faster rate with s than the QD case, but the magnitude of increment is still not significant.

Table 1: simulation result for coalescence time of $M/M/c$ queue

(QD: $\lambda_s = s, c_s = 1.2s, \mu = 1$)

s	mean	95% confidence interval
100	6.4212	[6.2902, 6.5522]
500	7.0641	[6.9848, 7.1434]
1000	7.7465	[7.6667, 7.8263]

Table 2: simulation result for coalescence time of $M/M/c$ queue

(QED: $\lambda_s = s, c_s = s + 2\sqrt{s}, \mu = 1$)

s	mean	95% confidence interval
100	6.5074	[6.3771, 6.6377]
500	8.5896	[8.4361, 8.7431]
1000	9.4723	[9.3041, 9.6405]

Finally we run a numerical experiment aiming to test how computational complexity of our algorithm changes with traffic intensity, $\rho = \lambda/c\mu$. Here we define the computational complexity as the total number of renewals (including arrivals and services/vacations) the algorithm samples in total to find the coalescence time. We expect the complexity to scale like $(c+1)(1-\rho)^{-2}E[T(\rho)]$ where $(c+1)$ is the number of renewal processes we need to simulate, $(1-\rho)^{-2}$ is on average the amount of renewals we need to sample to find its running time maximum for each renewal process, and $E[T(\rho)]$ is the mean coalescence time when the traffic intensity is ρ. Table 3 summarizes our numeral results, based 5000 independent runs of the algorithm for each ρ. We run the coalescence check at 10×2^k for $k = 1, 2, \ldots,$ until we find the coalescence. We observe that as ρ increase, the computational complexity increases significantly, but when multiplied by $(1-\rho)^2$, the resulting products are of about the same magnitude - up to a factor proportional to λ, given that the number of arrivals scales as *lambda* per unit time. Therefore, the main scaling parameter for the complexity here is $(1-\rho)^{-2}$. Notice that if we simulate the system forward in time from empty, it also took around $O\left((1-\rho)^{-2}\right)$ arrivals to get close to stationary.

References

[1] S. Asmussen. *Applied Probability and Queues.* Springer, 2 edition, 2003.

[2] S. Asmussen, P. Glynn, and H. Thorisson. Stationarity detection in the initial transient problem. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 2(2):130–157, 1992.

[3] J. Blanchet and X. Chen. Steady-state simulation of reflected brownian motion and related stochastic networks. *arXiv preprint arXiv:1202.2062*, 2013.

Table 3: simulation result for computational complexities with varying traffic intensities

$M/M/c$ queue with fixed $\mu = 5$ and $c = 2$

λ	traffic intensity (ρ)	mean number of renewals sampled	mean index of successful inspection time	mean number of renewals sampled $\times (1-\rho)^2$
5	0.5	225.6670	11.7780	56.4168
6	0.6	377.0050	14.7780	60.3208
7	0.7	764.3714	21.9800	68.7934
8	0.8	2181.3452	44.2320	87.2538
9	0.9	12162.6158	161.0840	121.6262

[4] J. Blanchet and J. Dong. Perfect sampling for infinite server and loss systems. Forthcoming in Advances in Applied Probability, 2014.

[5] J. Blanchet and A. Wallwater. Exact sampling fot the steady-state waiting times of a heavy-tailed single server queue. arXiv:1403.8117v1.pdf, 2014.

[6] H. Chen and D.D. Yao. *Fundamentals of queueing networks: Performance, asymptotics, and optimization*, volume 46. Springer Science & Business Media, 2013.

[7] S.B. Connor and W.S. Kendall. Perfect simulation for a class of positive recurrent markov chains. *The Annals of Applied Probability*, 17(3):781–808, 06 2007.

[8] S.B. Connor and W.S. Kendall. Perfect simulation of M/G/c queues. arXiv:1402.7248v1, 2014.

[9] J.N. Corcoran and R.L. Tweedie. Perfect sampling of ergodic harris chains. *The Annals of Applied Probability*, 11(2):438–451, 05 2001.

[10] K. Ensor and P. Glynn. Simulating the maximum of a random walk. *Journal of Statistical Planning and Inference*, 85:127–135, 2000.

[11] S.G. Foss and R.L. Tweedie. Perfect simulation and backward coupling. *Stochastic Models*, 14:187–203, 1998.

[12] D. Garmarnik and D. Goldberg. Steady-state GI/GI/n queue in the Halfin-Whitt regime. *Annals of Applied Probability*, 23:2382–2419, 2013.

[13] F.P. Kelly. *Reversibility and stochastic networks*, volume 40. Wiley, 1979.

[14] W. Kendall. Geometric ergodicity and perfect simulation. *Electron. Comm. Probab.*, 9:140–151, 2004.

[15] W. Kendall and J. Møller. Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes. *Advances in Applied Probability*, pages 844–865, 2000.

[16] W.S. Kendall. Perfect simulation for the area-interaction point process. In *Probability towards 2000*, pages 218–234. Springer, 1998.

[17] J. Propp and D. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[18] R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo method*, volume 707. John Wiley & Sons, 2011.

[19] K. Sigman. Exact simulation of the stationary distribution of the FIFO M/G/c queue. *Journal of Applied Probability*, 48A:209–216, 2011.

[20] K. Sigman. Exact sampling of the stationary distribution of the FIFO M/G/c queue: the general case for $\rho < c$. *Queueing Systems*, 70:37–43, 2012.

[21] W. Whitt. Multiple channel queues in heavy traffic I. *Advances in Applied Probability*, 2(1):150–177, 1970.